Received: 7 April 2023 | *Revised:* 24 May 2023 | *Accepted:* 28 May 2023
DOI: 10.37917/ijeee.20.1.2 *Vol. 20 | Issue 1 | June 2024*

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access
IJEEE University of Basrah College of Engineering

An Assessment of Ensemble Voting Approaches, Random Forest, and Decision Tree Techniques in Detecting Distributed Denial of Service (DDoS) Attacks

Mustafa S. Ibrahim Alsumaidaie *[1], Khattab M. Ali Alheeti [2], Abdul Kareem Alaloosy [2]
[1]Department of Computer Science, University of Anbar, Ramadi, Iraq
[2]Computer Networking Systems Dept., University of Anbar, Ramadi, Iraq

Correspondance
*Mustafa S. Ibrahim Alsumaidaie
Department of Computer Science,
University of Anbar, Ramadi, Iraq
Email: mustafa.s.alsomadae@uoanbar.edu.iq

Abstract
The reliance on networks and systems has grown rapidly in contemporary times, leading to increased vulnerability to cyber assaults. The Distributed Denial-of-Service (Distributed Denial of Service) attack, a threat that can cause great financial liabilities and reputation damage. To address this problem, Machine Learning (ML) algorithms have gained huge attention, enabling the detection and prevention of DDOS (Distributed Denial of Service) Attacks. In this study, we proposed a novel security mechanism to avoid Distributed Denial of Service attacks. Using an ensemble learning methodology aims to it also can differentiate between normal network traffic and the malicious flood of Distributed Denial of Service attack traffic. The study also evaluates the performance of two well-known ML algorithms, namely, the decision tree and random forest, which were used to execute the proposed method. Tree in defending against Distributed Denial of Service (DDoS) attacks. We test the models using a publicly available dataset called TIME SERIES DATASET FOR DISTRIBUTED DENIAL OF SERVICE ATTACK DETECTION. We compare the performance of models using a list of evaluation metrics developing the Model. This step involves fetching the data, preprocessing it, and splitting it into training and testing subgroups, model selection, and validation. When applied to a database of nearly 11,000 time series; in some cases, the proposed approach manifested promising results and reached an Accuracy (ACC) of up to 100 % in the dataset. Ultimately, this proposed method detects and mitigates distributed denial of service. The solution to securing communication systems from this increasing cyber threat is this: preventing attacks from being successful.

Keywords
Ensemble Voting, Cybersecurity, Decision Tree, Random Forest, Anomaly Detection, DDoS Attack.

I. Introduction

The World Wide online was introduced more than 20 years ago. Since then, it has become a significant worldwide force that profoundly impacts daily life through a wide range of online applications that provide billions of web pages daily [1]. Web applications are now asynchronous, interactive, and dynamic. They may be found in a wide range of contexts. Because of its crucial global importance, it is now essential to ensure that web applications are accurate, safe, and of the highest caliber [2]. Among the most important issues facing today's digital environment are distributed denial of service (DDoS) and denial of service (DoS) attacks. This grave danger, which has existed since the internet's founding, presents a difficult situation that cannot yet be resolved by using the TCP/IP protocol as it currently stands. Making the system unworkable is the primary objective of distributed denial of service attacks. Despite notable advances in information security technology, these risks continue to evaluate the

effectiveness of the defences that are in place [3]. Integrating devices with sensors, software, processing power, networking capabilities, and other technologies to enable them to operate independently while connecting and interacting with other systems and devices over the Internet is known as the Internet of Things, or IoT. This development intends to improve and simplify services in a variety of industries. In these fields, the Internet of Things has the potential to improve sensing skills and maximize resource usage. However, given the sensitive nature of the data these devices create and networks' general proliferation and scope, there are serious privacy and security risks in many IoT applications. Moreover, difficulties with energy economy, computing capacity restrictions, and device memory limitations exacerbate security problems and design difficulties. To guarantee safe data transfer across a myriad of connected devices, it is imperative to establish protocols and security mechanisms that satisfy the changing demands of both present and future devices. Attacks known as distributed denial of service (DDoS) pose a severe danger to system security [4].

A DDoS attack happens when several computers overload the bandwidth or resources of a targeted system, such as a web server or server, according to the Cisco Internet Report for 2018–2023. This kind of attack usually happens when a number of compromised systems send a large amount of traffic to the target system. Distributed denial of service attacks are considered by most service providers to be their main risk. More than half of operators report suffering infrastructure outages, indicating that they are still an issue. Attackers with amplification capabilities who possess tools for Distributed Denial of Service attacks take advantage of weaknesses in networks and computer resources. The security sector is working hard to make these assaults unprofitable for online thieves.

The scale and frequency of Distributed Denial of Service attacks are on the rise:

• When compared to years, the greatest attack volume increased by 63%.

• The frequency of Denial of Service (DoS) assaults increased by 39% globally in the past year.

• Approximately 23% of these assaults were more than 1 Gbps.

• One Gbps or more was the size of 23% of the assaults.

• Distributed denial of service attacks have an average size of 1 Gbps and have the ability to completely destroy companies.

Figure 1 shows the number of DDoS assaults is expected to quadruple to 15.4 million worldwide by 2023, according to projections.

Because ML algorithms can evaluate massive quantities of data and find patterns that might indicate an assault, they are

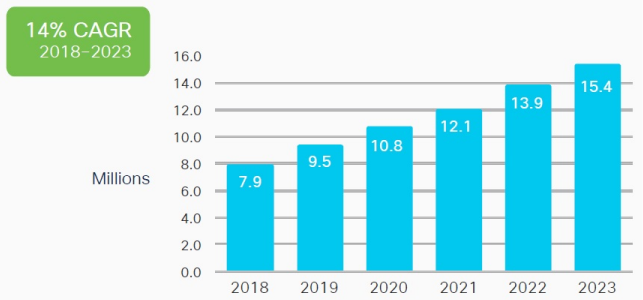

Fig. 1. The Number of DDoS attacks: Attacks will double to 15.4 million by 2023 globally [4]

frequently employed to detect DDoS attacks. Among the most effective ML algorithms for DDoS detection installations are Random Forest and Decision Tree. This decision-making model is based on the decision tree, which is an inexpensive and simple algorithm. In contrast, Random Forest is an ensemble learning algorithm as it is trained on several decision trees and then combines their output to provide results that are more reliable and accurate. [5]. This study's primary goal is to assess the effectiveness of Random Forest and decision tree algorithms for DDOS assault classification. We used a dataset of network traffic types that is available to the public to conduct tests with these models. The remaining portion of this study is organized as follows: section 2 presents the relevant work on employing ML techniques to identify Distributed Denial of Service threats. The format of the paper is as follows: in Section 3, we describe our dataset and approach. The findings are shown in Section 4. Random Forest algorithms and decision trees are evaluated for performance. The paper is concluded in portion 5, the last portion.

II. Literature Review

The prevalence of Distributed Denial of Service (DDoS) assaults has increased recently, leading to the creation of a number of detection and mitigating techniques. In this field, ML techniques have been shown to be useful. These algorithms are renowned for their capacity to analyze huge datasets and spot trends that could point to an intrusion. This section reviews previous work on ML algorithms for DDS attack detection. The studies included range from 2018 to 2022, as shown below:

To detect Denial of Service attacks, Abdulrahman & Ibrahem [6] put up a novel architecture for a host-based intrusion detection system. IDS uses a dynamic, recurring evaluation of intruder groups in relation to the entities around them to enable intelligent intrusion detection. The CICIDS 2017 dataset, which includes network traffic subject to both benign and Distributed Denial of Service attacks and meets certifiable

requirements, was used to assess the effectiveness of the suggested approach. To find the best attributes for identifying particular kinds of targeted assaults, we thoroughly examined a number of ML approaches and network traffic patterns. The results of the investigation show that the RF and C5.0 classifiers perform better than other classifiers, with average ACC of 0.868 and 0.8654, respectively. Moreover, the ACC rate of these classifiers is around 0.99. [6] , suggesting a high probability of success.

Bindraa & Sooda [7] described a method for choosing the best-supervised ML algorithm to identify DDS Attacks in their work. They trained these algorithms using real-world datasets in order to evaluate their ACC as well. There were two main questions that motivated the research: What is the best effective supervised learning method for identifying denial-of-service attacks? Moreover, to what extent do these algorithms demonstrate PRE (PRE) when trained on real-world data? Our study's findings show that the Random Forest Classifier achieved an astounding 96% ACC rate. Two distinct measurements corroborate this finding. In addition, our research revealed that the LR approach yielded an ACC of 82%, the KNN methodology provided an ACC of 94%, and the RF method created an ACC of 96%. [7] .

Using the UNBS-NB 15 and KDD99 datasets, Tuan et al. [8] carried out an empirical investigation on ML techniques for Botnet Distributed Denial of Service (DDoS) attack detection. This analysis looked at a number of ML techniques. Results reveal that the KDD99 dataset performed better than the UNBS-NB 15 dataset. The experimental results of detecting Botnet DDoS attack are the further validation of ML methods in computer security and other fields.

A highly advanced technique for identifying Distributed DDoS attacks within a network was showcased. by G. Usha et al. [9]. Their system uses several ML methods. Furthermore, a deep learning architecture based on Convolutional Neural Networks (CNNs) is integrated to identify and classify attacks. According to the results, out of all the methodologies that were assessed, the XGBoost algorithm gets the greatest ACC. Table 1 offers a synopsis of relevant research.

The paper's literature study demonstrates a thorough investigation of ML methods for identifying Distributed Denial of Service attacks. Nonetheless, a number of research gaps remain apparent. First, rather than post-attack analysis, real-time detection capabilities—which are essential for proactive threat mitigation—are not given enough attention. Furthermore, the majority of standardized datasets are used to assess the performance of these detection systems, which leaves a vacuum in our knowledge of how well they function in a variety of real-world network settings with varying traffic patterns. Inadequate attention is also paid to the integration of these

TABLE I.
SUMMARIZATION LITERATURE REVIEW

Ref.	Algorithms	ACC
[7]	Random Forest, SVM.	%86.80, %79.88.
[8]	Random Forest, SVM, KNN	%96.13, %82.35 %94.36.
[9]	Decision Tree, Random Forest.	%77, %86.
[10]	KNN, XGBoost.	%87, %89.
[6]	SVM	%84.32

detection models into current cybersecurity infrastructures, an aspect that is essential to their useful use in real-world operating environments.

Moreover, although the algorithms' effectiveness is sometimes discussed, the scalability and computing requirements of these systems as networks grow are rarely examined. The models' capacity to adjust to changing Distributed Denial of Service methods and vectors—which are dynamic and constantly becoming more sophisticated—is another crucial overlook. Finally, despite its rising frequency and the difficulties it presents in extracting valuable characteristics for attack detection, the effect of increased traffic encryption on the effectiveness of these models is not adequately investigated. Filling up these gaps will greatly improve ML models' applicability and efficacy in identifying distributed denial of service attacks in a variety of dynamic and varied scenarios.

In order to identify distributed denial of service attacks, a new method is presented in this research, which uses a time series dataset that performs faster and more effectively than current approaches. An overview of earlier research using ML approaches for Distributed Denial of Service attack detection carried out between 2018 and 2022, is attached. These works cover deep learning approaches, supervised learning algorithms, ML experiments, and host-based intrusion detection systems. Even though these approaches have shown encouraging results, the topic of this research makes use of a quicker time series dataset, which speeds up attack detection and eventually increases the suggested approach's overall efficacy.
The paper's literature study demonstrates a thorough investigation of ML methods for identifying distributed denial of service attacks. Nonetheless, a number of research gaps remain apparent. First, real-time detection capabilities—which are more important for proactive threat prevention than for post-attack analysis—are not given enough attention. Furthermore, standardized datasets are typically used to assess the effectiveness of these detection systems, which leaves a vac-

uum in our knowledge of how well they function in a variety of real-world network contexts with varying traffic patterns. Inadequate attention is also paid to the integration of these detection models into current cybersecurity infrastructures, an aspect that is essential to their useful use in operational environments.

Moreover, while the algorithms' efficiency is sometimes discussed, little research has been done on how these systems would scale in terms of processing requirements as networks grow. The models' capacity to adjust to changing Distributed Denial of Service (DDoS) tactics and vectors, which are becoming more sophisticated, is another crucial overlook. Despite its rising frequency and the difficulties it presents for feature extraction in attack detection, the effect of increased traffic encryption on the effectiveness of these models is also not thoroughly investigated. Closing these gaps will greatly improve ML models' applicability and efficacy in identifying DDoS assaults in a variety of dynamic and varied scenarios. Using a time series dataset that beats current techniques in speed and effectiveness, this research presents a unique method for identifying DDoS attacks. Included is an overview of research done between 2018 and 2022 that employed ML methods to identify DDoS attacks. These investigations include deep learning methodologies, supervised learning algorithms, ML experiments, and host-based intrusion detection systems. Although these approaches have shown encouraging results, the methodology used in this study makes use of a quicker time series dataset, which improves detection speed and overall efficacy.

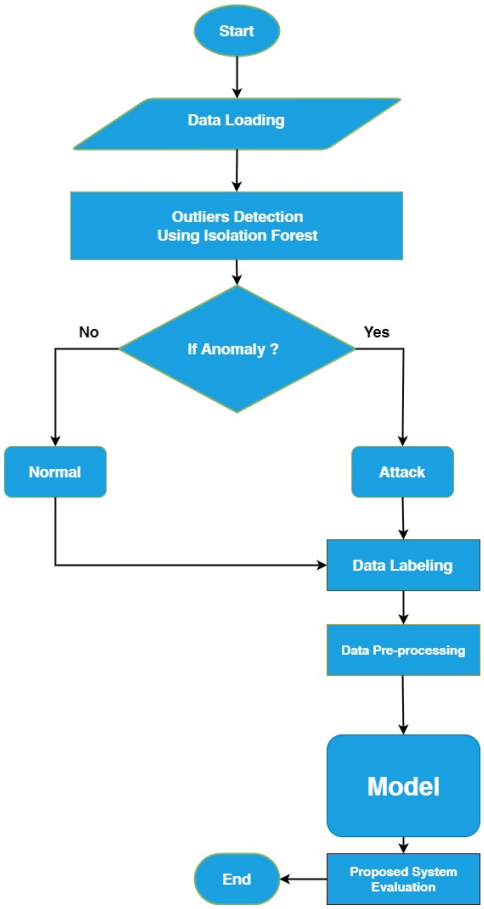

Fig. 2. The Flow Chart of the System

III. Methods

There are the steps to MLs Step 1: Get the relevant information regarding the issue. Following this, appropriate attributes are selected and the data is prepared. Then, apply an ML tool to generate a model specific to your problem domain. The model is then analyzed for the reliability lastly on the evaluation stage the ACC and productivity of a model while forecasting the results. This system employs supervised ML algorithms, namely, Decision Tree Classifier (DT), Random Forest (RF), and Ensemble Voting to address the problem.

A. The General Structure of The System

The architecture of the proposed system and the implementation steps are shown in Fig 2.

B. Time Series Dataset Description

CICDoS2019 is a time series dataset, first introduced by the well-known Canadian Institute for Cybersecurity (CIC) [11]. S. Ratan Kumar et al. presented the CICDoS 2019 dataset in 2021. For distant assaults, Cornoir employed a detector. Their goal was to devise a method that would enable time series data to be quickly evaluated using parallel processing, providing an early warning system against flooding attacks that cause a denial of service assault. They were able to detect the attacks more swiftly as a result of their increased ability to produce time series data. The researchers split the CICDoS2019 benchmark dataset into four comparable time series datasets for the purpose of identifying TCP-level flooding assaults. According to their research, the new technique could manage 2.3 times as much attack traffic as sequential processing techniques. The CICDoS2019 dataset includes PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP among these modern reflecting DDoS assaults. The training session began at 10:30 on January 12 and finished at 17:15 on the same day as the exam, which began at 09:40 on March 11 and ended at 17:35. As shown in Fig 3, the dataset consists of 4 characteristics and 5893 records.

The Table II below provides a brief description of each feature in the dataset, allowing for a better understanding of

	File number	time_period	#SYN Packets	#SYN-ACK Packets	#ACK Packets	#RST Packets	#TCP Packets
0	1	1	5	4	38	4	47
1	1	2	1	2	21	4	24
2	1	3	1	0	41	2	42
3	1	4	13	10	91	0	114
4	1	5	88	90	2068	24	2246
...	...	...	...	...	...	...	...
5526	818	5527	36516	4	5817	3763	46100
5527	818	5528	39516	33	1817	1763	43129
5528	818	5529	31516	5	2897	2763	37181
5529	818	5530	37516	12	3817	4763	46108
5530	818	5531	37516	34	3223	3456	44229

Fig. 3. Screenshot of a time series dataset.

TABLE II.
TIME SERIES DATASET DESCRIPTION

Feature	Description
Pcap file number	The number assigned to the Pcap file
Time period serial number	The serial number assigned to the time period
Number of SYN packets	The count of packets with the SYN flag set
Number of SYN-ACK packets	The count of packets with both SYN and ACK flags set
Number of ACK packets	The count of packets with the ACK flag set
Number of RESET packets	The count of packets with the RESET flag set
Number of TCP packets	The total count of TCP packets

the information they represent.

C. Dataset Pre-Processing

Pre-processing the dataset is the next stage in creating an ML model. This step is crucial as it involves modifying and preparing the data. The goal of data preparation is to reduce the amount of information. Pre-processing is a method used to prepare data for analysis, as raw data is of limited value for analysis. To prevent overfitting, we performed pre-processing on the data before feeding it into a classification system. Pre-processing must be applied to build a prediction model with accurate results, which is one of the essential steps. Normalization is applied in the preprocessing stage; it scales the features to fit in a common range. In this way, the model's performance is consistent and stable. There are many scaling techniques, such as standard scaler and min-max scaler. Results of Our Study: All features were scaled to a common scale and min-max scaler.RIGHT_LOWER.

1) Dataset Balancing Using Over-Sampling Technique

Reading the dataset we found, the dataset has a different class imbalance, most of them are majority classes. On the other hand, the rest of the classes are the minority class classes with a small number of samples to learn from, this class imbalance influences the efficiency of the ML system to opt out conclusions correctly. Since the dataset classes are imbalanced, an algorithm called SMOTE (Synthetic Minority Over-sampling Technique) was used to pre-process and balance the dataset in Python environment, as demonstrated in Fig. 4.

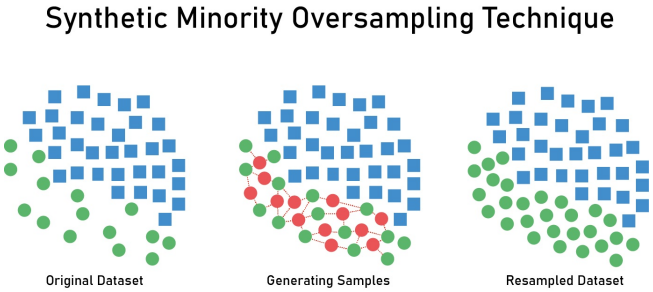

Fig. 4. SMOTE (Synthetic Minority Over-sampling Technique)

Steps to Balancing Dataset using Synthetic Minority Over-sampling Technique (SMOTE):

1. Start Process — flowchart initiates
2. Load Dataset: Load the dataset you want to balance to memory.
3. Imbalance Check — Before training any model, first check if the class label is imbalanced.
4. The above code will not solve the above-mentioned problem if we are dealing with an imbalanced data set.
5. If a dataset is not imbalanced, then keep the SMOTE oversampling code and proceed with the original dataset only.
6. Process as SMOTE: Use the SMOTE algorithm, which is a type of oversampling method to randomly generate synthetic samples(synthetic images) of the data for the minority class.
7. Over-sample Minority Class: Combine the original minority class samples and the synthetic samples generated by SMOTE to over-sample the minority class.
8. Save Balanced Dataset: The balanced dataset, i.e. original majority class samples combined with newly generated minority class samples.
9. End Process: This is the end of flowchart.

SMOTE for balancing a dataset: Fig Fig. 5 represents the flowchart of SMOTE process.

D. Prediction Models

The fourth part in building a model to solve a problem is applying one of ML techniques. Ensemble Voting with two

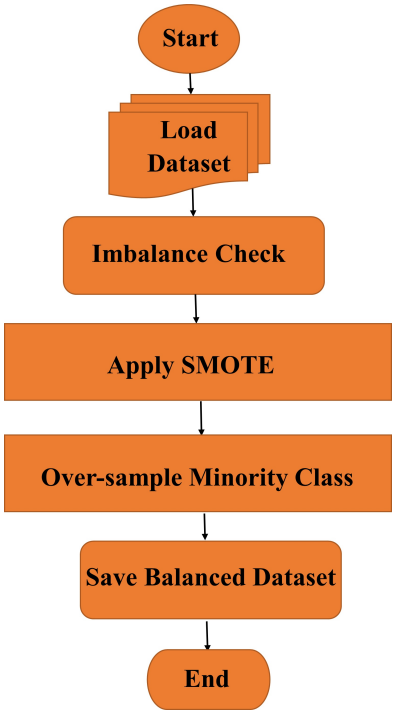

Fig. 5. Dataset Balancing Using Over-Sampling Technique.

single ML algorithms (RF and DT) for Distributed Denial of Service attack prediction using time series data View File.

1) Ensemble Learning Techniques

The goal of ensemble learning, a recent development in the field of AI and data mining, is to combine many learning methods to increase classifier computational complexity while also improving prediction ACC and performance. Ensembles are complex ML methods that are often utilized when the primary interest lies in prediction ACC or PRE rather than in having an easily interpretable model [12]. This paper presents the ensemble learning algorithm voting [13].

E. Voting

Voting Classifier: Voting Classifier is an ensemble learning classifier that combines similar and different scores of a set of classifiers and delivers the final output from it based on majority voting. It harmonises the outputs from a number of multiple classifiers, but rather than a simple majority vote, a voting system is used to decide the most accurate prediction. This approach is able to combine various voting schemes, similar to hard voting and soft voting which have a specific way of aggregating probabilistic classification predictions from the classifiers. A detailed description of these techniques will be given in the next section [13].

1) Hard Voting

This process is nothing but the majority voting, which calculates mode value and According to the counting (like which label is maximum) consider that label as voted winner. Not only did using the SVM and KNN algorithms give unsatisfactory results for DDoS attack detection, but I also used an ensemble approach to time series and Ensemble Learning techniques. It means combining many ML models to make it more strong and accurate. The first ensemble was built from SVM, KNN and RF algorithms named model1, model2 and model3 with the name Ensemble1. This voting-based approach (a high voting mechanism) finally verified the last prediction of the majority vote on these models. Model-wise training with individual models followed by fusion drastically improved the detection system ACC while reducing the number of false positives and negatives. The flowchart in Fig. 6 The system's data flow from training data to individual models, to ensembles, and ultimately to the output is depicted, providing a visual representation of the ML process as it is explained.

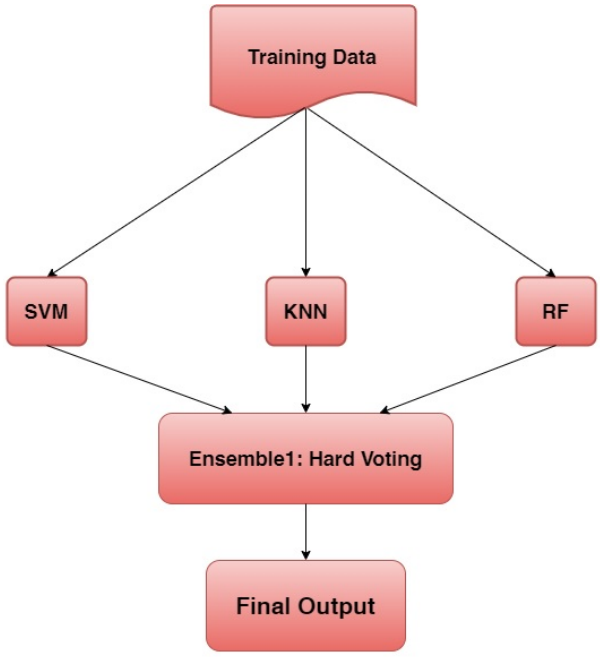

Fig. 6. Ensemble Flowchart

2) Random Forest (RF)

One of the most prevalent ML-based classification models is known as the Decision Tree (DT). This model is composed of a multitude of DTs, each of which can make a decision based on specific parameters. The DT is made up of two kinds of

nodes: parent and child nodes. The topmost node is referred to as the root node, while the end nodes are known as leaf nodes. These leaf nodes are responsible for making decisions based on given conditions. However, the DT is highly sensitive to any changes in the data. Any modification made to the data, without altering the conditions, may lead to erroneous decisions by the DT. To address this issue, the Random Forest (RF) model was introduced. The RF consists of several DTs, each of which can independently make decisions based on randomized datasets. These datasets are created by altering the data occurrences without changing the data length. This way, each tree receives equally sized data and the problem of sensitivity to data changes is mitigated [14]. To elevate the ACC of the output, it is essential that the correlation between the trees remains low. A high correlation between the trees would result in the propagation of inaccurate decisions, thereby decreasing the overall PRE of the algorithm. The final decision is made based on the majority vote of all the trees [15].

3) Decision Tree (DT)

The decision tree is a commonly used supervised ML algorithm; it can visualize with an inverse tree structure to represent a particular decision problem. Decision trees are classified into two types: Tree classification (CT) and Tree Regression (RT). Nowadays, decision tree algorithms are called CART; this term refers to classification and regression trees. CART is the name Leo Breiman uses to refer to decision tree algorithms that are used to build models to solve classification and regression problems. The most significant difficulty in implementing a Decision Tree is identifying the attribute that constitutes the root node and each level. Information gain and Gini index are used to select the attribute that may be designated the root node at each level [16]. the pseudo-code below for implementing Random Forest (RF) and Decision Tree (DT) algorithms for DDoS attack detection using a time series dataset:

BEGIN

-IMPORT LIBRARIES

-LOAD THE DATASET

-PRE-PROCESS THE DATASET

-SPLIT THE DATASET INTO FEATURES (X) AND THE TARGET VARIABLE (Y)

-SPLIT THE DATASET INTO TRAINING AND TESTING SETS

-TRAIN A DECISION TREE MODEL

-MAKE PREDICTIONS USING THE DECISION TREE MODEL

-CALCULATE THE ACCURACY OF THE DECISION TREE MODEL

-TRAIN A RANDOM FOREST MODEL

-MAKE PREDICTIONS USING THE RANDOM FOREST MODEL

-CALCULATE THE ACCURACY OF THE RANDOM FOREST MODEL

-COMPARE THE ACCURACIES OF THE DECISION TREE AND RANDOM FOREST MODELS

END

IV. Evaluation Measures and Results

11,423 occurrences of the IDDOSAD technique were used in the evaluation. The prediction model's ACC was evaluated using a variety of assessment indicators. The following equations demonstrate how ACC, F1-score (F1-S), Pre, Recall (REC), and Training Time were used to assess how well supervised ML detected DDoS assaults [17]:

ACC This is the ratio of correct predictions (both positive and negative) to all predictions.

$$ACC = \frac{(tp+tn)}{(tp+tn+fp+fn)} \tag{1}$$

F1-S: This is the harmonic mean of PRE and REC, balancing these two metrics. It ranges from 0 to 1 and is particularly useful for imbalanced classes.

$$F1-S = 2 \times \frac{(Pre \times REC)}{(Pre+REC)} \tag{2}$$

PRE): PRE is the ratio of correctly predicted positive observations to the total predicted positive observations. High PRE indicates a small number of False Positives.

$$Prec = \frac{tp}{(tp+fp)} \tag{3}$$

REC: Also known as Sensitivity or True Positive Rate, REC is the ratio of correctly predicted positive observations to all observations in the actual class. High REC indicates a small number of False Negatives [18].

$$REC = \frac{tp}{(tp+fn)} \tag{4}$$

As indicated in Table 3, the following outcomes were attained:

With 100% ACC, F1-s, Prec, and Rec, RF was the model that performed the best. Training took 3.54 seconds, indicating a rather quick learning curve.

DT: The DT model also performed remarkably well, obtaining 100% Rec, F1-s, ACC, and Prec. This model has the quickest training time of all the evaluated models, at just 0.06 seconds.

Ensemble 1: Acc, F1-s, and Rec of the first ensemble model were 92%, with a slightly higher Prec of 93%. But at

6.77 seconds, this model's training time was the longest of all the models.

Finally, it can be said that the RF and DT models both performed well in identifying and thwarting DDoS assaults, providing a dependable defence for communication systems against this expanding cybersecurity risk. The ensemble models also produced encouraging results, suggesting that integrating several methods as demonstrated in Table III.

After calculating the execution time, DT takes 0.06 seconds to execute, which is the lowest execution time (Training time) compared to other algorithms, as shown in Fig. 7.

Fig. 7. Evaluating each ML algorithm's training time in comparison.

The comparison findings show that DT is the most effective algorithm for categorizing DDOS assaults in the suggested methodology.

V. Conclusion

This paper has systematically evaluated the efficacy of ensemble voting approaches, the Random Forest (RF) algorithm, and the Decision Tree (DT) technique in the context of Distributed Denial of Service (Distributed Denial of Service) attack detection. The integration of these methods within a ML framework have been shown to significantly enhance the detection capabilities for cybersecurity threats, particularly in handling large volumes of network traffic data associated with the Distributed Denial of Service attacks.

TIME SERIES DATASET FOR DISTRIBUTED DENIAL OF SERVICE ATTACK DETECTION is the dataset which provides all the necessary attributes which are required for evaluating the performance metrics like ACC, PRE, REC, and F1 score, which is performed during the study. Both the RF and DT models demonstrated good results on all the metrics measured, which reveals their consistency and effectiveness. From the results, it can be seen that the ensemble model had an ACC of only 92%, but that is still okay, and the advantage of the ensemble model is to take the best of all models you have trained and used for prediction. The research also covers the rapidity of the Decision Tree Model in the training sessions, thus making it perfect for real-time responsive applications. On the other hand, in spite of being a bit time-consuming to train, the Random Forest model showed to be a powerful tool to detect DDoS attacks, thanks to its ensemble nature, allowing it to adapt to different network behaviours and attack patterns. At the end of the study, it was discovered that ML algorithms— Random Forest, Decision Trees, Outlier Detection, Principal Component Analysis, Linear Discrimination Analysis, etc, were very efficient in the identification of Distributed Denial of Service attacks. Our approaches not only score high on ACC benchmarks, but they also show deployability in practice as a result of fast training times and robustness against various data characteristics. Future work might include integrating these models with streaming live data and employing ensemble methods to enhance detection performance and resilience to sophisticated cyber threats. Moreover, broadening the dataset to cover emerging Distributed Denial of Service attack types may enhance the model's efficiency and readiness for evolving cybersecurity challenges.

Acknowledgment

I am utterly grateful to Allah Ta'ala for giving me the ability to do this. I would like to express my sincere thanks to my supervisor, Dr. Khattab M. Ali Alheeti, for his advice and for suggesting the topic of this study. I am grateful to him for his insightful advice, insightful criticism, and helpful recommendations throughout the research.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

References

[1] Y.-F. Li, P. K. Das, and D. L. Dowe, "Two decades of web application testing—a survey of recent advances," *Information Systems*, vol. 43, pp. 20–54, 2014.

[2] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From shallow to deep learning," *arXiv preprint arXiv:2008.00364*, 2020.

[3] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[4] R. Hummel, C. Hildebrand, H. Modi, and G. Sockrider, "Netscout threat intelligence report," *Netscout Systems, Inc., Tech. Rep*, 2020.

TABLE III.
COMPARING THE OUTCOMES.

Model	ACC (%)	F1-S (%)	PRE (%)	REC (%)	Training time (sec)
RF	100%	100%	100%	100%	3.54
DT	100%	100%	100%	100%	0.06
Ensemble 1	92%	92%	93%	92%	6.77

[5] S. Wani, M. Imthiyas, H. Almohamedh, K. Alhamed, S. Almotairi, and Y. Gulzar, "Distributed denial of service (ddos) mitigation using blockchain—a comprehensive insight. symmetry 2021, 13, 227," 2021.

[6] A. A. Najar and S. Manohar Naik, "Ddos attack detection using mlp and random forest algorithms," *International Journal of Information Technology*, vol. 14, no. 5, pp. 2317–2327, 2022.

[7] A. A. Abdulrahman and M. K. Ibrahem, "Evaluation of ddos attacks detection in a new intrusion dataset based on classification algorithms," *Iraqi Journal of Information and Communication Technology*, vol. 1, no. 3, pp. 49–55, 2018.

[8] N. Bindra and M. Sood, "Detecting ddos attacks using machine learning techniques and contemporary intrusion detection dataset," *Automatic Control and Computer Sciences*, vol. 53, pp. 419–428, 2019.

[9] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of botnet ddos attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, pp. 283–294, 2020.

[10] G. Usha, M. Narang, and A. Kumar, "Detection and classification of distributed dos attacks using machine learning," in *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020*, pp. 985–1000, Springer, 2021.

[11] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–8, IEEE, 2019.

[12] M. Jabbar, R. Aluvalu, *et al.*, "Rfaode: A novel ensemble intrusion detection system," *Procedia computer science*, vol. 115, pp. 226–234, 2017.

[13] C. Hu, *Ensemble feature learning-based event classification for cyber-physical security of the smart grid*. PhD thesis, Concordia University, 2019.

[14] O. Rahman, M. A. G. Quraishi, and C.-H. Lung, "Ddos attacks detection and mitigation in sdn using machine learning," in *2019 IEEE world congress on services (SERVICES)*, vol. 2642, pp. 184–189, IEEE, 2019.

[15] S. S. Priya, M. Sivaram, D. Yuvaraj, and A. Jayanthiladevi, "Machine learning based ddos detection," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 234–237, IEEE, 2020.

[16] Á. Michelana, J. Aveleira-Mata, E. Jove, H. Alaiz-Moretón, H. Quintián, and J. L. Calvo-Rolle, "Denial of service attack detection based on feature extraction and supervised techniques," in *International Symposium on Distributed Computing and Artificial Intelligence*, pp. 59–68, Springer, 2022.

[17] M. S. I. Alsumaidaie, K. M. A. Alheeti, and A. K. Al-Aloosy, "Intelligent detection system for a distributed denial-of-service (ddos) attack based on time series," in *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 445–450, IEEE, 2023.

[18] M. M. AL-Ani, N. Omar, and A. A. Nafea, "A hybrid method of long short-term memory and auto-encoder architectures for sarcasm detection," *J. Comput. Sci*, vol. 17, no. 11, pp. 1093–1098, 2021.